ECE363 Assignment 4

Student ID:

1. Consider transferring an enormous file of L bytes from Host A to Host B. Assume an MSS of 1,460 bytes.

   (a) What is the maximum value of L such that TCP sequence numbers are not exhausted? Recall that the TCP sequence number field has 4 bytes.

   TCP sequence number is in terms of bytes.

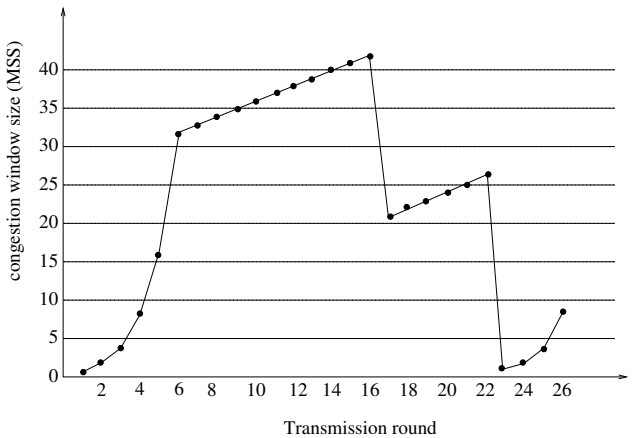
   $L = 2^{32} = 4294967296$ bytes.

   (b) For the L you obtain in (a), find how long it takes to transmit the file. Assume that a total of 66 bytes of transport, network, and data-link header are added to each segment before the resulting packet is sent out over a 10 Mbps link. Ignore flow control and congestion control so A can pump out the segments back to back and continuously.

   Each segment has 1460 bytes payload plus 66 bytes overheads.

   $2^{32} \times \frac{1460+66}{1460} \times 8/(10^7) \approx 3591.3$ seconds.

2. Assuming TCP Reno is the protocol experiencing the behavior shown below. Answer the following questions. Provide a short discussion (be brief) justifying your answer.



(a) Identify the intervals of time when TCP slow start is operating.

Round 1- 6, and 23 - 26. The congestion window size is exponentially increased during the slowstart stage.

(b) Identify the intervals of time when TCP congestion avoidance is operating.

Round 6 - 16, 17 - 22. The congestion window size is linearly increased during the congestion avoidance stage.

(c) After the 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

By triple duplicate ACK, because the congestion window size is halved after a triple duplicate ACK.

(d) After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

By a timeout, because the congestion window size is reduced to one after a time out.

(e) What is the initial value of slowstart threshold at the first transmission round?

32 MSS. When the congestion window is smaller than 32, the window is exponentially increased; when the congestion window exceeds 32, it is linearly increased.

(f) What is the value of slowstart threshold at the 18th transmission round?

21 MSS, because the slowstart threshold equals the half of the window size when the packet loss is detected.

(g) What is the value of slowstart threshold at the 24th transmission round?

13 MSS, because the slowstart threshold equals the half of the window size when the packet loss is detected.

(h) During what transmission round is the 70th segment sent?

7. During the first six round, $1 + 2 + 4 + 8 + 16 + 32 = 63$ segments are sent; during the seventh round, another 33 segments are sent.

(i) Assuming a packet loss is detected after the 26th round by the receipt of a triple duplicate ACK, what will be the values of the congestion window size and of the slowstart threshold?

4 MSS. The congestion window size and the slowstart threshold are both equal to the half of the window size at 26th round.

3. We note that TCP waits until it has received three duplicate ACKs before performing a fast retransmit. Why do you think the TCP designers chose not to perform a fast retransmit after the first duplicate ACK for a segment is received?

   Since the IP networks do not provide any guarantee reliable and in-order packet delivery services, a single duplicate ACK may be due to packet re-order, or packet duplication, instead of packet loss. Fast retransmission after the first duplicate ACK may not be necessary and may waste network resources (because TCP un-necessarily reduces its cwnd after non-congestion events).

4. Optional question: A TCP sender sends an enormous file over a link between routers A and B, with 10 Mbps. No other traffic share the link with it. Assume that there is no transmission error of the link, and the routers' buffer size is much smaller than the file size. Is it possible that there is no packet lost for the whole file transfer? If so, how?

   Yes, it's possible. Since the sender's window size is the minimal of the congestion window size (cwnd) and the receiver advertised window size (rwnd), we can set the rwnd low enough to avoid overshooting the link bandwidth, so no congestion loss will happen.